RECEIVED
CENTRAL FAX CENTER

SEP 2 8 2006

AMENDMENTS TO THE CLAIMS

1. (Currently Amended) A method, comprising:

examining information about branch instructions that reach a write-back stage of processing within a processor;

defining a plurality of streams based on the examining, wherein each stream comprises a sequence of basic blocks in which only a last block in the sequence ends in a branch instruction, the execution of which causes program flow to branch, the remaining basic blocks in the stream each ending in a branch instruction, the execution of which does not cause program flow to branch;

switching to a stream mode if a branch-to-mesocode transition is detected in one of the plurality of streams; and

switching to a normal mode if a mesocode-to-branch transition is detected.

2. (Original) The method of claim 1, further comprising storing identifying information for each defined stream.

3. (Original) The method of claim 2, wherein the identifying information comprises a start instruction pointer and an end instruction pointer for each stream.

4. (Original) The method of claim 1, further comprising collecting dependent information for each stream, the dependent information identifying a dependent stream, being a child stream that is executed after the stream during an instance of program execution, and the dependent information also indicating a probability of the dependent stream being executed after the stream.

5. (Original) The method of claim 4, further comprising predicting a target stream or block that is likely to be executed based on a current instruction pointer.

6. (Original) The method of claim 1, further comprising storing at least some of the basic blocks within a stream in contiguous memory locations.

7. (Previously Presented) The method of claim 1, further comprising converting at least some of the instructions in a stream into ISA-implementation specific instructions, and storing the ISA-implementation specific instructions in memory locations contiguous to the basic blocks.

8. (Currently Amended) The method of claim 1, further comprising switching the processor between ~~a~~ the stream mode in which instructions from a stream are prefetched based on a prediction and ~~a~~ the normal mode in which instructions within a basic block are fetched based on the prediction.

9. (Currently Amended) A processor, comprising:

a mechanism to examine information about branch instructions that reach a write-back stage of processing within the processor; ~~and~~

a mechanism to define a plurality of streams based on the examining, wherein each stream comprises a sequence of basic blocks in which only a last block in the sequence ends in a branch instruction, the execution of which causes program flow to branch, the remaining basic blocks in the stream each ending in a branch instruction, the execution of which does not cause program flow to branch;

a mechanism to switch to a stream mode if a branch-to-mesocode transition is detected in

one of the plurality of streams; and

a mechanism to switch to a normal mode if a mesocode-to-branch transition is detected.

10. (Original) The method of claim 9, further comprising a mechanism to store identifying information for each defined stream.

11. (Original) The processor of claim 10, wherein the identifying information comprises a start instruction pointer and an end instruction pointer for each stream.

12. (Original) The processor of claim 9, wherein the mechanism to define the plurality of streams further collects dependent information for each stream, the dependent information identifying a dependent stream being a child stream that is executed after the stream during an instance of program execution, the dependent information also indicating a probability of the dependent stream being executed after the stream.

13. (Original) The processor of claim 12, wherein the mechanism to define the plurality of streams further comprises a prediction mechanism to predict a target stream or block that is likely to be executed based on a current instruction pointer.

14. (Original) The processor of claim 9, further comprising storing at least some of the basic blocks within a stream in contiguous memory locations.

15. (Previously Presented) The processor of claim 9, further comprising converting at least some of the instructions in a stream into ISA-implementation specific instructions, and storing

the ISA-implementation specific instsructions in memory locations contiguous to the basic blocks.

16. (Currently Amended) The processor of claim 9, further comprising a mechanism to switch the processor between ~~a~~ the stream mode in which instructions from a stream are prefetched based on a prediction and ~~a~~ the normal mode in which instructions within a basic block are fetched based on the prediction.

17. (Currently Amended) A system, comprising:

a processor comprising a mechanism to examine information about branch instructions that reach a write-back stage of processing within the processor, ~~and~~ a mechanism to define a plurality of streams based on the examining, wherein each stream comprises a sequence of basic blocks in which only a last block in the sequences ends in a branch instruction, the execution of which causes program flow to branch, the remaining basic blocks in a stream each ending in a branch instruction, the execution of which does not cause program flow to branch, a mechanism to switch to a stream mode if a branch-to-mesocode transition is detected in one of the plurality of streams, and a mechanism to switch to a normal mode if a mesocode-to-branch transition is detected; and

a memory coupled to the processor.

18. (Original) The system of claim 17, wherein the processor further comprises a mechanism to store identifying information for each defined stream.

19. (Original) The system of claim 18, wherein the identifying information comprises a start

instruction pointer and an end instruction pointer for each stream.

20. (Currently Amended) A processor, comprising:

a fetch/prefetch unit;

a branch prediction unit to supply a branch target address of a predicted branch based on a current instruction pointer to the fetch/prefetch unit; ~~and~~

a stream prediction unit to supply a stream target address of a predicted stream based on a current instruction pointer to the fetch/prefetch unit, wherein the predicted stream comprises a sequence of basic blocks in which only a last block in the sequence ends in a branch instruction, the execution of which causes program flow to branch, the remaining basic blocks in the stream each ending in a branch instruction, the execution of which does not cause program flow to branch;

a mechanism to switch to a stream mode if a branch-to-mesocode transition is detected in one of the plurality of streams; and

a mechanism to switch to a normal mode if a mesocode-to-branch transition is detected.

21. (Original) The processor of claim 20, wherein the stream prediction unit further comprises a mechanism to define a plurality of streams based on an examination of information about branch instructions that reach a write-back stage of processing within the processor.

22. (Original) The processor of claim 21, wherein the stream prediction unit further comprises a mechanism to store identifying information for each defined stream.

23. (Original) The processor of claim 22, wherein the identifying information comprises a

start instruction pointer and an end instruction pointer for each stream.

24. (Original) The processor of claim 20, wherein the stream prediction unit comprises a mechanism that collects dependent information for each stream, the dependent information identifying a dependent stream being a child stream that is executed after the stream during an instance of program execution, the dependent information also indicating a probability of the dependent stream being executed after the stream.

25. (Original) The processor of claim 20, further comprising a mechanism to store at least some of the basic blocks within a stream in contiguous memory locations.

26. (Previously Presented) The processor of claim 20, further comprising a mechanism to convert at least some of the instructions in a stream into ISA-implementation specific instructions, and to store the ISA-implementation specific instructions in contiguous memory locations contiguous to the basic blocks.

27. (Currently Amended) A system, comprising:

a processor comprising a fetch/prefetch unit, a branch prediction unit to supply a branch target address of a predicted branch based on a current instruction pointer to the fetch/prefetch unit, ~~and~~ a stream prediction unit to supply a stream target address of a predicted stream based on a current instruction pointer to the fetch/prefetch unit, wherein the predicted stream comprises a sequence of basic blocks in which only a last block in the sequence ends in a branch instruction, the execution of which causes program flow to branch, the remaining basic blocks in the stream each ending in a branch instruction, the execution of which does not cause program flow to

branch, a mechanism to switch to a stream mode if a branch-to-mesocode transition is detected in one of the plurality of streams, and a mechanism to switch to a normal mode if a mesocode-to-branch transition is detected; and

a memory coupled to the processor.

28. (Original) The system of claim 27, wherein the stream prediction unit further comprises a mechanism to define a plurality of streams based on an examination of information about branch instructions that reach a write-back stage of processing within the processor.

29. (Original) The system of claim 28, wherein the stream prediction unit further comprises a mechanism to store identifying information for each defined stream.

30. (Original) The system of claim 29, wherein the identifying information comprises a start instruction pointer and an end instruction pointer for each stream.